Date=05/10/2020
Lecture By=Arkesh Jaiswal
Subject ⇒Javascript-2

IN PREVIOUS LECTURE **(QUICK RECAP) Date-29/09/2020**	**In Today's Lecture (Overview)**
➢ Javascript ➢ Why Study JavaScript? ➢ JavaScript Syntax ➢ JavaScript Values ➢ JavaScript Literals ➢ JavaScript Variables ➢ JavaScript Operators ➢ JavaScript Keywords ➢ JavaScript Comments ➢ JavaScript Operators ➢ JavaScript Arithmetic Operators ➢ JavaScript Assignment Operators ➢ JavaScript Comparison Operators ➢ MCQs ➢ Questions for self practice / CC for the day ➢ JavaScript Reference Book	JavaScript Conditional Statements The JavaScript Switch Statement The break Keyword The default Keyword JavaScript Loops Different Kinds of Loops The For Loop JavaScript While Loop JavaScript Break and Continue Questions For Self Practice

JavaScript Conditional Statements

Very often when you write code, you want to perform different actions for different decisions.

You can use conditional statements in your code to do this.

In JavaScript we have the following conditional statements:

- Use `if` to specify a block of code to be executed, if a specified condition is true

- Use `else` to specify a block of code to be executed, if the same condition is false
- Use `else if` to specify a new condition to test, if the first condition is false
- Use `switch` to specify many alternative blocks of code to be executed

---

The if Statement

Use the `if` statement to specify a block of JavaScript code to be executed if a condition is true.

Syntax

```
if (condition) {
  //  block of code to be executed if the condition is true
}
```

Note that `if` is in lowercase letters. Uppercase letters (If or IF) will generate a JavaScript error.

Example

Make a "Good day" greeting if the hour is less than 18:00:

```
if (hour < 18) {

  greeting = "Good day";}
```

The result of greeting will be:

```
Good day
```

The else Statement

Use the `else` statement to specify a block of code to be executed if the condition is false.

```
if (condition) {
  //  block of code to be executed if the condition is true
} else {
  //  block of code to be executed if the condition is false
}
```

Example

If the hour is less than 18, create a "Good day" greeting, otherwise "Good evening":

```
if (hour < 18) {

  greeting = "Good day";

} else {

  greeting = "Good evening";

}
```

The result of greeting will be:

```
Good day
```

The else if Statement

Use the `else if` statement to specify a new condition if the first condition is false.

Syntax

```
if (condition1) {
  //  block of code to be executed if condition1 is true
} else if (condition2) {
  //  block of code to be executed if the condition1 is false and
condition2 is true
} else {
  //  block of code to be executed if the condition1 is false and
condition2 is false
}
```

Example

If time is less than 10:00, create a "Good morning" greeting, if not, but time is less than 20:00, create a "Good day" greeting, otherwise a "Good evening":

```
if (time < 10) {

  greeting = "Good morning";

} else if (time < 20) {

  greeting = "Good day";

} else {

  greeting = "Good evening";

}
```

The result of greeting will be:

```
Good day
```

The JavaScript Switch Statement

Use the `switch` statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {
```

```
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

This is how it works:

- **The switch expression is evaluated once.**
- **The value of the expression is compared with the values of each case.**
- **If there is a match, the associated block of code is executed.**
- **If there is no match, the default code block is executed.**

Example

The `getDay()` method returns the weekday as a number between 0 and 6.

(Sunday=0, Monday=1, Tuesday=2 ..)

This example uses the weekday number to calculate the weekday name:

```
switch (new Date().getDay()) {
```

```
case 0:
  day = "Sunday";
  break;
case 1:
  day = "Monday";
  break;
case 2:
   day = "Tuesday";
  break;
case 3:
  day = "Wednesday";
  break;
case 4:
  day = "Thursday";
  break;
case 5:
  day = "Friday";
  break;
case 6:
  day = "Saturday";
```

```
}
```

The result of day will be:

```
Monday
```

The break Keyword

When JavaScript reaches a `break` keyword, it breaks out of the switch block.

This will stop the execution of inside the block.

It is not necessary to break the last case in a switch block. The block breaks (ends) there anyway.

Note: If you omit the break statement, the next case will be executed even if the evaluation does not match the case.

The default Keyword

The `default` keyword specifies the code to run if there is no case match:

Example

The getDay() method returns the weekday as a number between 0 and 6.

If today is neither Saturday (6) nor Sunday (0), write a default message:

```
switch (new Date().getDay()) {
```

```
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
    break;
  default:
    text = "Looking forward to the Weekend";
}
```

The result of text will be:
Looking forward to the Weekend

JavaScript Loops

Loops are handy, if you want to run the same code over and over again, each time with a different value.

Often this is the case when working with arrays:

Instead of writing:

```
text += cars[0] + "<br>";
text += cars[1] + "<br>";
text += cars[2] + "<br>";
```

```
text += cars[3] + "<br>";

text += cars[4] + "<br>";

text += cars[5] + "<br>";
```

You can write:

```
var i;

for (i = 0; i < cars.length; i++) {

  text += cars[i] + "<br>";

}
```

Different Kinds of Loops

JavaScript supports different kinds of loops:

- `for` **- loops through a block of code a number of times**
- `for/in` **- loops through the properties of an object**
- `for/of` **- loops through the values of an iterable object**
- `while` **- loops through a block of code while a specified condition is true**
- `do/while` **- also loops through a block of code while a specified condition is true**

The For Loop

The `for` loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {

  // code block to be executed

}
```

JavaScript While Loop

Loops can execute a block of code as long as a specified condition is true.

The While Loop

The `while` loop loops through a block of code as long as a specified condition is true.

Syntax

```
while (condition) {

  // code block to be executed

}
```

Example

In the following example, the code in the loop will run, over and over again, as long as a variable (i) is less than 10:

Example

```
while (i < 10) {
  text += "The number is " + i;
  i++;
}
```

If you forget to increase the variable used in the condition, the loop will never end. This will crash your browser.

The Do/While Loop

The `do/while` loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

Syntax

```
do {
  // code block to be executed
}
while (condition);
```

Example

The example below uses a `do/while` loop. The loop will always be executed at least once, even if the condition is false, because the code block is executed before the condition is tested:

Example

```
do {
  text += "The number is " + i;
  i++;
}
while (i < 10);
```

JavaScript Break and Continue

The `break` statement "jumps out" of a loop.

The `continue` statement "jumps over" one iteration in the loop.

The Break Statement

You have already seen the `break` statement used in an earlier chapter of this tutorial. It was used to "jump out" of a `switch()` statement.

The `break` statement can also be used to jump out of a loop.

The `break` statement breaks the loop and continues executing the code after the loop (if any):

Example

```
for (i = 0; i < 10; i++) {

  if (i === 3) { break; }

  text += "The number is " + i + "<br>";

}
```

The Continue Statement

The `continue` statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

This example skips the value of 3:

Example

```
for (i = 0; i < 10; i++) {

  if (i === 3) { continue; }

  text += "The number is " + i + "<br>";

}
```

Questions For Self Practice

1.https://au-assignment.s3.ap-south-1.amazonaws.com/JS-hw-b02e9788-bfe4-4291-a2a1-5d3a6a63cd92.pdf
2.https://au-assignment.s3.amazonaws.com/JS-cc-05f06143-9586-4d26-a9be-68a2b78a7957.pdf